Ora devo verificare che effettivamente p_k è una direzione di decrescita:

$$(\nabla\phi(x))^T p_k = -r_k^T(r_k+\beta_k p_{k-1}) =$$
$$\stackrel{!}{=} -r_k^T r_k - \beta_k r_k^T p_{k-1} =$$

(viene perché abbiamo scelto α_k, non dipende nemmeno da p_k)

Ora dimostreremo che $r_k^T p_{k-1}$ è uguale a zero e quindi

$$\vdots = -\|r_k\|_2^2 < 0 \quad \forall k$$

PROP: 1) $\forall k \quad r_{k+1}^T p_k = 0$ (vale in generale)

2) $\forall k \quad r_{k+1}^T r_k = 0$ quindi a coppie i residui sono ortogonali (in senso euclideo) (vale solo per CG)

OSS: $r_{k+1} := b - Ax_{k+1} = b - A(x_k+\alpha_k p_k) = (b-Ax_k) - \alpha_k A p_k =$

$$\stackrel{!}{=} r_k - \alpha_k A p_k$$

dim: (1) $r_{k+1}^T p_k = (r_k - \alpha_k A p_k)^T p_k = r_k^T p_k - \alpha_k p_k^T A p_k = 0$ (PROP)

def di $\alpha_k = \frac{r_k^T p_k}{p_k^T A p_k}$

(2) $r_{k+1}^T r_k = r_{k+1}^T (p_k - \beta_k p_{k-1}) = \cancel{r_{k+1}^T p_k}^{=0 \text{ per (1)}} - \beta_k r_{k+1}^T p_{k-1} =$

$$\stackrel{!}{=} -\beta_k (r_k - \alpha_k A p_k)^T p_{k-1} = -\beta_k \cancel{r_k^T p_{k-1}} + \alpha_k \beta_k p_k^T A p_{k-1} = 0$$

perché $p_k \perp_A p_{k-1}$ per def dei p_k. ■

Ora vogliamo vedere l'algoritmo del CG. Prima di scrivere l'algoritmo vediamo delle RISCRITTURE di α_k e β_k per RIDURRE IL COSTO COMPUTAZIONALE. Abbiamo

$$\alpha_k := \frac{r_k^T p_k}{p_k A p_k} = \frac{r_k^T r_k}{p_k A p_k}$$

NB non è che $p_k = r_k$ ma quando faccio il prodotto con r_k^T vengono uguali

$$\beta_k := \frac{-r_k^T A p_{k-1}}{p_{k-1}^T A p_{k-1}} = \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}$$

← sono spariti i prodotti matriciali che hanno costo computazionale maggiore. In più $r_{k-1}^T r_{k-1}$ l'ho già calcolato al passo precedente.

dim: • $r_k^T p_k = r_k^T(r_k + \beta_k p_{k-1}) = r_k^T r_k + \beta_k \cancel{r_k^T p_{k-1}}^{=0 \text{ per (1)}} = r_k^T r_k$ ← vale in generale, perché (1) vale in generale, mi piace perché sto riscrivendo α_k, che non "dipende" dal metodo CG

• $p_k^T r_{k-1} \stackrel{(OSS)}{=} p_k^T(r_k + \alpha_{k-1} A p_{k-1}) = p_k^T r_k + \alpha_{k-1} \cancel{p_k^T A p_{k-1}}^{=0} =$

$$\stackrel{!}{=} p_k^T r_k = r_k^T r_k$$

Ma ho anche:

$$p_k^T r_{k-1} = (r_k + \beta_k p_{k-1})^T r_{k-1} = \cancel{r_k^T r_{k-1}}^{=0 \text{ per (2)}} + \beta_k p_{k-1}^T r_{k-1} = \beta_k r_{k-1}^T r_{k-1} \quad \text{per } \bullet$$

Uguagliando le due quantità si ricava β_k. ■

ALGORITMO CG: pcg di Matlab, lavora in due modi: → A, b ⇝ x (eventi M precond) • Calcola Ax con prodotto matr.-vett. • Mz = r con LU

→ A, b (M) + function per fare i 2 pt.

Dato x_0 (in genere $x_0 = 0$), facciamo

1. $k=0$, $r_0 = b - Ax_0$ nella pratica non chiedo mai $r_k = 0$
2. Test d'arresto se "$r_k = 0$" stop
3. $\beta_k = \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}$ ($\beta_0 = 0$ se $k=0$)

$p_k = r_k + \beta_k p_{k-1}$

$\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}$

21/11/2012

$x_{k+1} = x_k + \alpha_k p_k$

$r_{k+1} = r_k - \alpha_k A p_k$ $(= b - A x_{k+1}$ ma nella pratica si usa l'altra)

4. $k = k+1$ e vai a 2.

Il costo peso nel metodo è il prodotto matrice vettore Ap_k che mi compare 2 volte (ma è sempre lo stesso) per come ho ottimizzato scrivendo α_k e β_k come ho fatto. $\Rightarrow O(n^2)$

Osservo che se avessi calcolato $r_{k+1} = b - Ax_{k+1}$ avrei avuto 2 prodotti matrice vettore diversi!

Poi ho dei prodotti scalari tra vettori che però costano meno le due tra l'altro si ripetono uguali).

Quindi il costo computazionale è un $O(n^2)$

OSS: È vero che in teoria se calcolo $r_{k+1} = b - Ax_{k+1}$ o con $r_k - \alpha_k A p_k$ è uguale ma in pratica con l'aritmetica floating point non essere uguale.
In pratica quando $k \approx \sqrt{n}$ se non ho ancora raggiunto la condizione d'arresto calcolo per una volta $r_{k+1} = b - Ax_{k+1}$ e poi riprendo a calcolarlo nell'altro modo. Così non incremento il costo comp. perché per una volta che lo faccio cambio poco, ma recupero precisione.

Il test d'arresto che si fa è quello del residuo relativo.

$$\frac{\|r_k\|_2}{\|b\|_2} = \frac{\|b - Ax_k\|_2}{\|(b - Ax_0)\|_{x_0=0}\|_2} < \varepsilon$$

($\|b\|_2$ lo calcolo una volta sola e quindi non incide sul costo)

Ora vediamo una proposizione che ci fa vedere dell'ortogonalità a coppie dei residui, all'ortogonalità tra tutti, in modo che così ho dimostrato che CG si conclude al più in n passi.

PROP: Sia $r_0 \neq 0$, $S \geq 1$ t.c. $r_k \neq 0 \ \forall k \leq S$ (i.e. se non ho già trovato la soluzione esatta)

$\Rightarrow \forall k, j = 0, \dots, S$, $k \neq j$ $\quad r_k \perp r_j$, $\quad p_k \perp_A p_j$

dim: Per induzione. • Prendo $S = 1$ $k = 1$ e $j = 0$

$r_1^T r_0 = (r_0 - \alpha_0 A p_0)^T r_0 = r_0^T r_0 - \alpha_0 p_0^T A r_0 = 0$ (ma qui lo sapevo già per la prop precedente)

$p_0^T A p_0 = 0$ per costruzione.

• Sia ora $S > 1$. hp di induzione: $\forall k, j = 0, \dots, S$, $k \neq j$ $\quad r_k^T r_j = 0$ e $p_k^T A p_j = 0$

Voglio dimostrare che:

$\forall j = 0, \dots, S-1$ $\quad r_{S+1}^T r_j = 0$ e $p_{S+1}^T A p_j = 0$

↳ tutto per $j = S$ è a posto per la PROP precedente.

$$r_{S+1}^T r_j = (r_S - \alpha_S A p_S)^T r_j = r_S^T r_j - \alpha_S p_S^T A r_j =$$

($r_S^T r_j = 0$ per hp di induzione)

$$\overset{!}{=} -\alpha_S p_S^T A (p_j - \beta_j p_{j-1}) = -\alpha_S p_S^T A p_j + \alpha_S \beta_S p_S^T A p_{j-1} = 0$$ per hp di induzione

$$p_{S+1}^T A p_j = (r_{S+1} + \beta_{S+1} p_S)^T A p_j = r_{S+1}^T A p_j + \beta_{S+1} p_S^T A p_j$$

($p_S^T A p_j = 0$ per hp di ind.)

Dato che ho $r_{k+1} = r_k - \alpha_k A p_k$ posso scrivere $A p_j = \alpha_j^{-1} (r_j - r_{j+1})$

$$\overset{!}{=} \alpha_j^{-1} (r_{S+1}^T r_j - r_{S+1}^T r_{j+1}) = 0$$ per la tesi già dimostrata. ∎

TEO: le CG applicato a $Ax=b$, $A \in \mathbb{R}^{n\times n}$ semidef. positiva, genera la soluzione esatta in al più n passi. (in aritmetica esatta)

dim: 7 solo n vettori $\perp$ tra loro. Ci mette di meno se la soluzione vive in un sottospazio di dim $<$

OSS: Abbiamo visto che se $x^* = A^{-1}b$ allora $\phi(x^*+e) = \phi(x^*) + \frac{1}{2} e^T A e$

se scriviamo $x = x^* + e \rightsquigarrow e = x - x^*$ possiamo riscrivere la relazione come:

$$\phi(x) = \phi(x^*) + \frac{1}{2}(x-x^*)^T A (x-x^*)$$

Minimizzare $\phi(x_k)$ equivale a minimizzare $\|e_k\|_A = \|A^{1/2} e\|_2$ ove $(A^{1/2} = Q \Lambda^{1/2} Q^H$ dove Q e Λ sono le matrici della fatt. $A = Q\Lambda Q^H$ agli autovalori.)

Quindi il metodo funziona anche se $A \in \mathbb{C}^{n\times n}$ se lo vediamo in questa chiave di lettura.

TEO: L'errore al passo $k+1$ è A-ortogonale a tutte le direzioni di ricerca precedenti:

i.e. $e_{k+1}^H A p_j = 0 \qquad \forall j = 0, \dots, k$

Inoltre: $\text{span}\langle p_0, p_1, \dots, p_k\rangle = \text{span}\langle Ae_0, \dots, A^{k+1}e_0\rangle$. (è SPAZIO di KRYLOV)

E quindi e_{k+1} è il vettore di norma A più piccola nello spazio $e_0 + \text{span}\langle Ae_0, \dots, A^{k+1}e_0\rangle$. (forse la dimostro la prossima volta o forse no)

Vediamo ora le STIME di VELOCITÀ di CONVERGENZA del CG

TEO:
$$\frac{\|e_k\|_A}{\|e_0\|_A} \le \min_{\substack{P_k \in \mathbb{P}_k \\ P_k(0)=1}} \left\{ \max_{i=1,\dots,n} |P_k(\lambda_i(A))| \right\}$$

(λ_i: autovalori di A; P_k: polinomi monici di grado k)

N.B.: Dipende da tutti gli autovalori della matrice A!

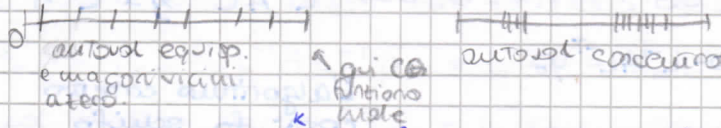
Dovremmo aver già visto il teo:

TEO:
$$\frac{\|e_k\|_A}{\|e_0\|_A} \le 2\left(\frac{\sqrt{K_2(A)}-1}{\sqrt{K_2(A)}+1}\right)^k \quad \text{con} \quad K_2(A) = \frac{\lambda_{max}(A)}{\lambda_{min}(A)}$$

la dimostrazione di questo risultato fa pesante uso di teoria dell'approssimazione (non è proprietà algebrica), e anche è fuorviante, perché ci fa vedere solo la dipendenza da λ_{max} e λ_{min}.

Questa seconda stima è pessimistica (e in pratica non mi dice niente, perché dice se A è mal condizionata potrebbe dare problemi.)

Il nostro Teorema invece è più fine, per esempio fa differenza tra i due casi:



dim: (TEO) $e_k = e_0 + \sum_{i=1}^{k} \gamma_i A^i e_0 = (1 + \sum \gamma_i A^i) e_0 = P_k(A) e_0$

con $P_k(x) = 1 + \sum \gamma_i x^i \in \mathbb{P}^k$ OSS: $P_k(0) = 1$!

Scrivo $A = Q\Lambda Q^H$ (s.d.p.)

$$\|e_k\|_A = \min_{\substack{P_k \in \mathbb{P}_k \\ P_k(0)=1}} \|A^{1/2} P_k(A) e_0\|_{A^2} = \min_{\substack{A_k \in P_k \\ P_k(0)=1}} \|P_k(A) A^{1/2} e_0\|_2 =$$

(posso far transitare la matrice a dx)

$$= \min_{//} \|Q P_k(\Lambda) Q^H A^{1/2} e_0\|_2$$

Osserviamo che il minimo dipende da e_0 e questo ci dà un po' noia, quindi facciamo una maggiorazione.

Si può mostrare che questa $\leq$ maggiorazione non è troppo poco fine (ci sono matrici per cui vale l'uguale).

$$\leq \min_{p} \|Q\, p_k(\Lambda)\, Q^H\|_2 \, \|A^{\frac{1}{2}} e_0\|_2 \overset{Q \text{ unitaria}}{=} \min_{p} \|p_k(\Lambda)\| \, \|e_0\|_A =$$

$$= \min_{p} \left\{ \max_{i=1,\dots,n} |\lambda_i(A)| \right\} \|e_0\|_A$$

Vediamo come ci comportiamo nel caso in cui la situazione sia brutta:

PRECONDIZIONAMENTO:

(1) $Ax = b \rightsquigarrow M^{-1}Ax = M^{-1}b$ con M opportuna (precondizionamento a sx)

Σ_A spettro di A (brutto)

Voglio che lo spettro di $M^{-1}A$ sia migliore per il metodo (di Krylov in generale).

OSS: Anche se in teoria con il precondizionamento mi porto a un problema ben condizionato, in pratica non sarà mai così, perché se A è mal condizionato non calcolerò bene il prodotto matrice-vettore $(M^{-1}A)$.

Quindi da un problema mal condizionato spero di ottenere un problema sempre mal condizionato ma che ha velocità di convergenza maggiore!

23/11/2012 (13)

(2) Analogamente è possibile fare il precondizionamento a dx:

$$Ax = b \rightsquigarrow AM^{-1}\hat{x} = b \quad \text{con } x = M^{-1}\hat{x}$$

(3) Per il CG si usa solitamente un precondizionamento simmetrico:

Scriviamo la fattorizzazione di Cholesky: $M = LL^T$ (uso un precond. sdp)

$$Ax = b \rightsquigarrow L^{-1}AL^{-T}y = L^{-1}b \quad \text{con } x = L^{-T}y$$

Come scegliamo M? Tipicamente si usa

M = fattorizzazione incompleta di Cholesky $\tilde{L}$

Perché vorrei una matrice "bella" della forma di A, detto che nei nostri problemi A è tridiagonale a blocchi vogliamo M e L fatte così. Lo fatt. di Cholesky riempie la matrice, l'incompleta invece lo fa come se fosse completa, ma lascia zero dove è zero.

M si sceglie in modo che $M^{-1}A$ ($o\ AM^{-1}$ $o\ L^{-1}AL^{-T}$) "approssimi" l'identità. In che senso approssima dipende dal metodo.

Inoltre chiaramente M deve essere t.c. risolvere $Mz = r$ sia più facile che risolvere $Ax = b$.

Vediamo come trasformiamo CG: $Ax = b \rightsquigarrow L^{-1}AL^{-T}y = L^{-1}b$

1. $k = 0 \quad \tilde{r}_0 = \cancel{b - Ax_0} = L^{-1}b - L^{-1}AL^{-T}y_0$
2. Se $\tilde{r}_k = 0$ mi fermo
3. $\tilde{\beta}_k = \dfrac{\tilde{r}_k^T \tilde{r}_k}{\tilde{r}_{k-1}^T \tilde{r}_{k-1}} \qquad \tilde{\beta}_0 = 0$

$$\tilde{p}_k = \tilde{r}_k + \tilde{\beta}_k \tilde{p}_{k-1}$$

$$\tilde{\alpha}_k = \frac{\tilde{r}_k^T \tilde{r}_k}{\cancel{p_k^T A p_k}} \qquad \tilde{p}_k^T L^{-1}AL^{-T}\tilde{p}_k$$

$$\cancel{x_{k+1} = x_k + \alpha_k p_k} \rightarrow y_{k+1} = y_k + \tilde{\alpha}_k \tilde{p}_k$$

$$\tilde{r}_{k+1} = \tilde{r}_k - \tilde{\alpha}_k \underbrace{\cancel{Ap_k}}_{L^{-1}AL^{-T}\tilde{p}_k}$$

L'algoritmo corretto così fatto scritto, ora vediamo effettivamente come si fa.

ALGORITMO PCG: su $Ax = b$

1. $k=0$ $\quad r_0 = b - Ax_0$
2. se $r_k = 0$ mi fermo
3. calcolo z_k t.c. $Mz_k = r_k$ ← l'unico aumento di costo computazionale sta qui, ovvero nella risoluzione di un sistema lineare in M ad ogni passo. (il resto costa uguale)

$$\tilde{\beta}_k = \frac{z_k^T r_k}{z_{k-1}^T r_{k-1}} \quad \text{* già calcolato al passo precedente}$$

$$p_k = z_k + \tilde{\beta}_k p_{k-1}$$

$$\tilde{\alpha}_k = \frac{z_k^T r_k}{p_k^T A p_k}$$

$$x_{k+1} = x_k + \tilde{\alpha}_k p_k$$

$$r_{k+1} = r_k - \tilde{\alpha}_k A p_k$$

$k = k+1$ e torno a 2.

N.B.: L'algoritmo è davvero il CG applicato al sistema precondizionato, è solo scritto in modo da diminuire il costo computazionale (ottimizzare)

Vediamolo:

$$\tilde{r}_k = L^{-1} b - L^{-1} A L^{-T} y_k = L^{-1}(b - A L^{-T} y_k) \overset{x_k = L^{-T} y_k}{=} L^{-1}(b - A x_k) = L^{-1} r_k$$

$$\tilde{\beta}_k = \frac{\tilde{r}_k^T \tilde{r}_k}{\tilde{r}_{k-1}^T \tilde{r}_{k-1}} = \frac{r_k^T L^{-T} L^{-1} r_k}{r_{k-1}^T L^{-T} L^{-1} r_{k-1}} \overset{M = LL^T}{=} \frac{r_k^T M^{-1} r_k}{r_{k-1}^T M^{-1} r_{k-1}} \overset{z_k = M^{-1} r_k}{=} \frac{z_k^T r_k}{z_{k-1}^T r_{k-1}}$$

Rimane da vedere che possiamo usare p_k al posto di $\tilde{p}_k$, ecc..

Abbiamo che $y = L^T x$ e quindi $\tilde{p}_k = L^T p_k$. (*)

$$p_k = L^{-T} \tilde{p}_k = L^{-T}(\tilde{r}_k + \tilde{\beta}_k \tilde{p}_{k-1}) = L^{-T}(L^{-1} r_k + \tilde{\beta}_k L^T p_{k-1}) = L^{-T} L^{-1} r_k + \tilde{\beta}_k p_{k-1} =$$

$$\overset{!}{=} M^{-1} r_k + \tilde{\beta}_k p_{k-1} = z_k + \tilde{\beta}_k p_{k-1} \quad \text{ok.}$$

$$\tilde{\alpha}_k = \frac{\tilde{r}_k^T \tilde{r}_k}{\tilde{p}_k^T L^{-1} A L^{-T} \tilde{p}_k} \overset{\text{visto per } \beta}{=} \frac{z_k^T r_k}{p_k^T A p_k} \to \text{per (*)}$$

$$y_{k+1} = y_k + \tilde{\alpha}_k \tilde{p}_k = L^T x_k + \tilde{\alpha}_k L^T p_k = L^T (x_k + \tilde{\alpha}_k p_k)$$

Ma ovviamente $y_{k+1} = L^T x_{k+1}$ e quindi uguagliando le due espressioni ottengo:

$$x_{k+1} = x_k + \tilde{\alpha}_k p_k$$

e tanto quanto posso fare la stessa cosa per r_{k+1}:

$$\tilde{r}_{k+1} = \tilde{r}_k - \tilde{\alpha}_k L^{-1} A L^{-T} \tilde{p}_k = L^{-1} r_k - \tilde{\alpha}_k L^{-1} A p_k = L^{-1}(r_k - \tilde{\alpha}_k A p_k)$$

ma d'altra parte $\tilde{r}_{k+1} = L^{-1} r_{k+1}$, da cui $r_{k+1} = r_k - \tilde{\alpha}_k A p_k$.

OSS: il costo del PCG è tipo: prodotto matrice-vettore + risoluzione sistema lineare in M ($Mz_k = r_k$)
con A (Ap_k)
= costo di CG

Il punto è, se uso CG ho $O(n^2)$ come costo comp., se dovessi mi toccasse fare n iterazioni arrivo a $O(n^3)$ ed è inutile, perché potrei usare gauss!

Se invece la matrice è sparsa CG mi costa $O(n)$ e anche la risoluzione di $Mz = r$ è $O(n)$ (se uso Cholesky incompleta), se ho ancora n iter. arrivo a $O(n^2)$ che è ancora sempre a Gauss per matrici sparse!

Quindi me la gioco sul numero di iterazioni! Il precondizionatore può fare davvero molto.

Vediamo ulteriori risultati sulla convergenza del metodo del gradiente coniugato. (dovuti a AXELSSON-LINDSCOG '86)

Consideriamo l'intervallo $[a,b]$, $a=\lambda_{min}$, $b=\lambda_{max}$ allora

$$p_k^*(\lambda) = T_k\left(\frac{b+a-2\lambda}{b-a}\right) \Big/ T_k\left(\frac{b+a}{b-a}\right)$$

(trasformazione affine $[-1,1] \to [a,b]$; normalizzazione per avere $p^*(0)=1$)

dove $p_k^*(\lambda)$ è il polinomio di miglior approssimazione del TEO di 2 lezioni fa (min max...) e T_k è il polinomio di Chebyshev di grado k:

$$T_k(x) = \cos(k\theta) \quad \text{con} \quad \theta = \arccos x$$
$$= \frac{1}{2}\left[(x+\sqrt{x^2-1})^k + (x-\sqrt{x^2-1})^k\right] \quad \text{su } [-1,1].$$

Abbiamo:

$$\max_{\lambda\in[a,b]} |p_k^*(\lambda)| = \frac{1}{T_k\left(\frac{b+a}{b-a}\right)} = \frac{2\sigma^k}{1+\sigma^{2k}} \quad \text{con} \quad \sigma = \frac{1-\sqrt{\frac{a}{b}}}{1+\sqrt{\frac{a}{b}}} = \frac{\sqrt{b}-\sqrt{a}}{\sqrt{b}+\sqrt{a}}$$

(continua)

Quindi se voglio che l'errore relativo sia $< \varepsilon$ (i.e. $\frac{\|e_k\|_A}{\|e_0\|_A} \le \varepsilon$) chiedo che

$$\frac{2\sigma^k}{1+\sigma^{2k}} \le \varepsilon$$

che facendo i conti diventa:

$$k \ge \left\lceil \log\left(\frac{1}{\varepsilon} + \sqrt{\frac{1}{\varepsilon^2} - 1}\right) \Big/ \log(\sigma^{-1}) \right\rceil$$

(→ si può trascurare perché se ε è piccolo $\frac{1}{\varepsilon^2}$ è grande)

$$\rightsquigarrow k \ge \left\lceil \frac{\log\left(\frac{2}{\varepsilon}\right)}{\log(\sigma^{-1})} \right\rceil = k^*(a,b,\varepsilon)$$

OSS: la prima cosa che noto è che la crescita rispetto a ε è logaritmica, e quindi ci piace.

Ora dobbiamo vedere cosa succede rispetto a σ^{-1}. Abbiamo:

$$\sigma^{-1} = \frac{\sqrt{b}+\sqrt{a}}{\sqrt{b}-\sqrt{a}} = \frac{\left(1+\sqrt{\frac{b}{a}}\right)}{\left(1-\sqrt{\frac{b}{a}}\right)} = \frac{\sqrt{K_2(A)}+1}{\sqrt{K_2(A)}-1} = 1 + \frac{2}{\sqrt{K_2(A)}-1}.$$

Pensiamo di avere una successione di sistemi lineari $A_n x_n = b_n$ (che poi è quello che abbiamo quando per esempio usiamo gli elem. finiti e facciamo tendere $h \to 0$), vogliamo vedere cosa succede per $n \to \infty$

Consideriamo per esempio gli el. finiti, abbiamo che

$$K_2(A_n) \nearrow \infty \quad \text{per } n \to \infty$$

e quindi $\sigma^{-1} = 1 + \varepsilon_n$ con $\varepsilon_n = \frac{2}{\sqrt{K_2(A_n)}-1} \searrow 0$

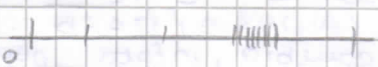
Dunque: $\log(\sigma^{-1}) = \log(1+\varepsilon_n) = \varepsilon_n + O(\varepsilon_n^2)$ e in conclusione

$$k^*(a,b,\varepsilon) = \left\lceil \frac{\log(2/\varepsilon)}{\varepsilon_n + O(\varepsilon_n^2)} \right\rceil =$$
$$= \left\lceil \log\left(\frac{2}{\varepsilon}\right) \varepsilon_n^{-1} (1+O(\varepsilon_n)) \right\rceil =$$
$$= \left\lceil \log\left(\frac{2}{\varepsilon}\right) \frac{\sqrt{K_2(A_n)}+1}{2} (1+O(\varepsilon_n)) \right\rceil$$

Quindi matematicamente stiamo dicendo che se $K_2(A)$ è alto ci vogliono più iterazioni, o meglio se $[a,b]$ è grande ci vogliono più iterazioni. (ci piacciono gli autovalori tutti concentrati e lontani da zero)

CONSIDERAZIONI FILOSOFICHE: Tipicamente la matrice che abbiamo noi (FEM/DF) avrà autovalori più o meno equispaziati:

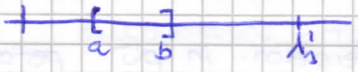
Il punto è trovare un precondizionatore che ci concentri gli autovalori e li porti in una configurazione tipo:



E gli autovalori isolati? Quelli in pratica li trascureremo usando un condizionamento essenziale (che tiene conto degli autoval concentrati) e poi mi occupo solo dopo degli outlier.

VARIAZIONI SUL TEMA: DISTRIBUZIONI di AUTOVALORI NON UNIFORMI

Pb: $\{\lambda_i(A)\} = \Sigma_A \subset [a,b]$ intervallo $\cup \{\lambda_1'\}$ $\quad b < \lambda_1'$



Il punto è che una volta sistemato il problema su $[a,b]$ con un iterazione sistemiamo anche λ_1'.

Abbiamo dal Teorema

$$\frac{\|e_k\|_A}{\|e_0\|_A} \leq \min_{\substack{p \in P_k \\ p(0)=1}} \max_{\lambda \in \Sigma_A} |p(\lambda)| .$$

Prendiamo $k = k^*(a,b,\varepsilon) + 1$ e un polinomio p particolare (che non è p^*)

$$\leq \max_{\lambda \in \Sigma_A} |p(\lambda)| . \quad (*)$$

Come scelgo p? Scelgo $p \in P_k = P_{k^*(a,b,\varepsilon)+1}$

$$p(\lambda) = p^*(\lambda) \cdot e(\lambda) \leftarrow e(\lambda) \text{ ad hoc : } e(\lambda) := 1 - \frac{\lambda}{\lambda_1'}$$

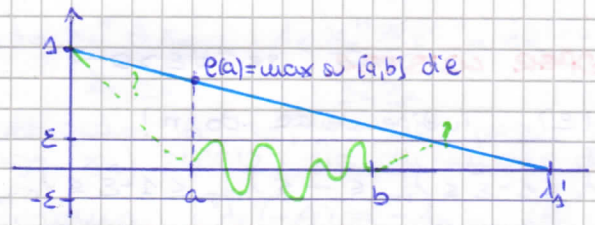
(p^*: polinomio di miglior approssimazione in $[a,b]$)

(osservo che $p(0) = p^*(0) \cdot e(0) = 1$ ok!)

Abbiamo:

$$(*) \leq \max \left\{ \underbrace{\max_{\lambda \in [a,b]} |p(\lambda)|}_{\text{nel continuo}} , \underbrace{|p(\lambda_1')|}_{0 \text{ per costruzione}} \right\} = \max_{\lambda \in [a,b]} |p(\lambda)|$$

Facciamo un disegno per capire la situazione:



$$0 < e(\lambda)|_{[a,b]} < e(a) = 1 - \frac{a}{\lambda_1'} < 1$$

Quindi abbiamo che:

$$\max_{\lambda \in [a,b]} |p(\lambda)| \leq \max_{\lambda \in [a,b]} |p^*(\lambda)| < \varepsilon$$

(perché $e(\lambda) < 1$ in $[a,b]$)

In conclusione ho fatto vedere che per aggiungere un autovalore fuori da $[a,b]$ non dobbiamo buttare via tutto, ma possiamo tenere conto di quello che abbiamo su $[a,b]$.

⊛ 1. $\Sigma_A \subset [a,b] \cup \left(\bigcup_{i=1}^{q} \lambda_i'\right)$ $\quad b < \lambda_i'$ (outlier a dx)

$\Rightarrow N(\varepsilon) = \underset{\#\text{ iter}}{K^*(a,b,\varepsilon)} + q$ Quindi q outlier a dx $\Rightarrow$ q iterazioni in più

⊛ 2. $\Sigma_A \subset [a,b] \cup \left(\bigcup_{i=1}^{p} \lambda_i\right)$ $\quad \lambda_i < a$ (outlier a sx)

$\Rightarrow N(\varepsilon) = K^*(a,b,\varepsilon^*) + p$ con $\varepsilon^* = \varepsilon \prod_{i=1}^{p} \frac{\lambda_i}{b}$

Quindi nel caso degli outlier a sx cambia, infatti, pensiamo di avere uno solo, usiamo

$e(\lambda) = 1 - \frac{\lambda}{\lambda_1}$ su $[a,b]$ ma $\left|1 - \frac{\lambda}{\lambda_1}\right| < \frac{b}{\lambda_1}$ NON di 1! non possiamo maggiorarla così

Se facessimo i conti come prima troveremmo:

$\max_{\lambda \in \Sigma_A} |p(\lambda)| < \max_{\lambda \in [a,b]} |p^*(\lambda)| \cdot \prod_{i=1}^{p} \frac{b}{\lambda_i} < \varepsilon$ sse $\max_{[a,b]} |p^*(\lambda)| < \varepsilon \prod_{i=1}^{p} \frac{\lambda_i}{b} = \varepsilon^*$.

Dato che $\varepsilon^* < \varepsilon$, non ho solo p iterazioni in più perché arrivo a che

$K^*(a,b,\varepsilon^*) > K^*(a,b,\varepsilon)$.

⊛ 3. $\Sigma_A \subset [a,b] \cup \left(\bigcup_{i=1}^{q} \lambda_i\right) \cup \left(\bigcup_{i=1}^{q} \lambda_i'\right)$ $\quad \lambda_i < a$, $\lambda_i' > b$ outlier

$\Rightarrow N(\varepsilon) = K^*(a,b,\tilde{\varepsilon}^*) + 2q$ con $\tilde{\varepsilon}^* = \varepsilon \prod_{i=1}^{q} 4 \frac{\lambda_i}{\lambda_i'} \left(1 - \frac{\lambda_i}{\lambda_i'}\right)^{-2} < \varepsilon$

(BRUTTO, ci interessa solo che è $< \varepsilon$)

Siano $\{A_n\}$ e $\{P_n\}$ succ. di matrici hermitiane.

def: Dico che c'è **SPECTRAL EQUIVALENCE** tra $\{A_n\}$ e $\{P_n\}$ se

$\lambda_i(P_n^{-1} A_n) \in [d_1, d_2] \quad \forall i \forall n$

con d_1, d_2 costanti positive indipendenti dalla dimensione.

Si dice anche che $\{P_n\}$ è **PRECONDIZIONATORE OTTIMALE** per $\{A_n\}$.

def: Dico che $\{A_n\}$ ha un **WEAK CLUSTER** (cluster debole) in p (valore) (es $p=1$) se $\forall \varepsilon > 0$, posto

$\gamma_{n,\lambda}(\varepsilon) = \#\{i : \lambda_i(A_n) \notin (p-\varepsilon, p+\varepsilon)\}$

ho $\lim_{n \to \infty} \frac{\gamma_{n,\lambda}(\varepsilon)}{n} = 0$. Ovvero ho pochi outlier rispetto alla dimensione

OSS: Se ho un cluster debole gli outlier possono aumentare ogni iterazione, ma in ogni caso la loro percentuale diminuisce (tende a zero, è inf).

def: Dico che ha un **PROPER CLUSTER** (cluster forte) se $\forall \varepsilon > 0$

$\gamma_{n,\lambda}(\varepsilon) \leq C(\varepsilon)$ indipendente da n!

⊛ 4. Σ_A t.c. $0 < \delta \leq \lambda_1 \leq \cdots \leq \lambda_i < 1 - \tilde{\varepsilon} \leq \lambda_{i+1} \leq \cdots \leq \lambda_{n-j} \leq 1 + \tilde{\varepsilon} \leq \lambda_{n-j+1} \leq \cdots \leq \lambda_n$

(quindi ho un cluster a 1, i outlier sx e j outlier dx).

$\Rightarrow \frac{\|e_k\|_A}{\|e_0\|_A} \leq 2 \left(\frac{1+\tilde{\varepsilon}}{\delta}\right)^i \tilde{\varepsilon}^{k-i-j}$ $\quad k > i + j = \#$ outlier.

28/11/2012 (15)

AUTOVALORI MATRICI HERMITIANE

def: **QUOZIENTE di RAYLEIGH** $r_A(x) = \frac{x^H A x}{x^H x}$ $\quad x \in \mathbb{C}^n \setminus \{0\}$, $A \in \mathbb{C}^{n \times n}$ hermitiana

MEMO: 1° $S \subseteq \mathbb{C}^n$ sottosp $\rightsquigarrow$ $\dim S^\perp = n - \dim S$

2° $S, T \subseteq \mathbb{C}^n$ stsp $\rightsquigarrow$ $\dim(S \cap T) \geq \max\{0, \dim S + \dim T - n\}$

3° $A \in \mathbb{C}^{m \times n}$ $m \leq n$ $S \subseteq \mathbb{C}^n$ stsp

$T = \{x \in \mathbb{C}^n : x = Ay \ y \in S\} = \mathrm{Im}_S(A)$

$N = \{x \in S : Ax = 0\} = \ker_S(A)$ $\rightarrow \dim T + \dim N = \dim S$

TEO: (COURANT-FISHER o minimax)
$A \in \mathbb{C}^{n \times n}$ Hermitiana con autovalori $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ allora

$$\lambda_{n-k+1}(A) = \min_{V_k} \max_{\substack{x \in V_k \\ x \neq 0}} r_A(x) \qquad \forall k = 1, \dots, n \qquad V_k \subseteq \mathbb{C}^n, \ \dim V_k = k$$

$$\lambda_k(A) = \max_{V_k} \min_{\substack{x \in V_k \\ x \neq 0}} r_A(x)$$

dim: $x_1, \dots, x_n$ autovettori ortonorm. relativi agli autovalori λ_i

$S := \langle x_k, \dots, x_n \rangle \rightarrow \dim S = n - k + 1$

Ma dal pto de del MEMO ho

$\dim(S \cap V_k) \geq \max\{0, (n-k+1) - k - n\} = 1$

$\rightarrow S \cap V_k \neq \{0\}$

Quindi posso prendere $x = \sum_{i=k}^{n} a_i x_i \in S \cap V_k$. Abbiamo: (posso prendere k perché sta in S)

$$r_A(x) = \frac{x^H A x}{x^H x} = \frac{\left(\sum_{i=k}^{n} \bar{a}_i x_i^H\right) A \left(\sum_{i=k}^{n} a_i x_i\right)}{\left(\quad\right)\left(\quad\right)}$$

(porto dentro la matrice e uso che $Ax_i = \lambda_i$ poi uso anche $x_i \perp x_j$ se $i \neq j$ quindi rimane una somma sola)

$$= \frac{\sum_{j=k}^{n} |a_j|^2 \lambda_j \underbrace{x_j^H x_j}_{=1}}{\sum_{j=k}^{n} |a_j|^2 \underbrace{x_j^H x_j}_{=1}} = \frac{\sum_{j=k}^{n} |a_j|^2 \lambda_j}{\sum_{j=k}^{n} |a_j|^2}$$

$$\leq \left(\max_{i=k-n} \lambda_i\right) \cdot \frac{\sum |a_j|^2}{\sum |a_j|^2} = \lambda_k$$

perché li avevo ordinati in ordine decrescente.

Quindi ho dimostrato che $r_A(x) \leq \lambda_k$, ma allora ovviamente

$$\min_{\substack{x \in V_k \\ x \neq 0}} r_A(x) \leq \lambda_k \qquad (*)$$

Ora considero x_k e scelgo $V_k = \langle x_1, \dots, x_k \rangle$

$$r_A(x_k) = \frac{x_k^H A x_k}{x_k^H x_k} = \lambda_k \frac{x_k^H x_k}{x_k^H x_k} = \lambda_k$$

e quindi abbiamo finito, (perché posso scegliere V_k in modo che contenga x_k) DUBBIO: ma dì uno dice che su V_k il minimo è x_k

Ora consideriamo $\lambda_k(-A) = -\lambda_{n-k+1}(A)$

$$\max_{V_k}\left(\min_{\substack{x \in V_k \\ x \neq 0}} \underbrace{r_{-A}(x)}_{-r_A(x)}\right) = \max_{V_k}\left(-\max_{\substack{x \in V_k \\ x \neq 0}} r_A(x)\right) =$$

$\min(-f) = -\max f$

$$\stackrel{!}{=} -\min_{V_k} \max_{x \in V_k} (r_A(x))$$

Conseguenza:

$\lambda_1 = \lambda_{max}(A) = \max_{x \neq 0} r_A(x)$ $\quad V_n = \mathbb{C}^n$ è l'unico sottospazio di dimensione n ($k = n$ nella 1ª)

$\lambda_n = \lambda_{min}(A) = \min_{x \neq 0} r_A(x)$

TEO: $A \in \mathbb{C}^{n \times n}$ Hermitiana, $U \in \mathbb{C}^{n \times n-1}$ t.c. $U^H U = I_{n-1}$, $B = U^H A U \in \mathbb{C}^{n-1 \times n-1}$. Allora

compressione di ordine 1

$$\lambda_1(A) \geq \lambda_1(B) \geq \lambda_2(A) \geq \dots \geq \lambda_{n-1}(A) \geq \lambda_{n-1}(B) \geq \lambda_n(A)$$

ovvero gli autovalori di B (che sono uno di meno) separano gli autovalori di A.

dim: Per prima cosa mostro che $\lambda_{k-1}(B) \geq \lambda_k(A)$ per $k = 2, \dots, n$

Uso la 2ª delle tesi del teorema di CF. So che $\exists\, Z_k \subseteq \mathbb{C}^n$, $\dim Z_k = k$ tale che

$$\lambda_k = \min_{\substack{x \in Z_k \\ x \neq 0}} r_A(x) \qquad \text{(i.e. } Z_k \text{ è lo spazio che realizza il max)}$$

Scrivo $U = [u_1 | \dots | u_{n-1}]$ e pongo $S = \langle u_1, \dots, u_{n-1} \rangle \rightsquigarrow \dim S = n-1$

$\Rightarrow \dim(Z_k \cap S) \geq \max\{0, \underbrace{\dim Z_k}_{k} + \underbrace{\dim S}_{n-1} - n\} = k-1 \geq 1$

Quindi $\exists\, x \in Z_k \cap S$ $x \neq 0$ t.c.

$$\lambda_k \leq \min_{\substack{x \in Z_k \cap S \\ x \neq 0}} r_A(x)$$

Ora $\forall x \in S \setminus \{0\}$ $\exists!\, y \in \mathbb{C}^{n-1} \setminus \{0\}$ t.c. $x = Uy$ $(y = U^H x)$

Consideriamo $W = \{y \in \mathbb{C}^{n-1} : y = U^H x \quad x \in Z_k \cap S\}$

$\dim W + \underbrace{\dim\{x \in S \cap Z_k : U^H x = 0\}}_{=0 \text{ per come è fatta } U} = \dim(S \cap Z_k) \geq k-1$

$\rightarrow \dim W \geq k-1$

Quindi abbiamo:

$$\lambda_k^{(A)} \leq \min_{\substack{x \in Z_k \cap S \\ x \neq 0}} r_A(x) = \min_{\substack{x \in Z_k \cap S \\ x \neq 0}} \frac{x^H A x}{x^H x} = \min_{\substack{y \in W \\ y \neq 0}} \frac{y^H U^H A U y}{y^H \underbrace{U^H U}_{= id_{n-1}} y} = \min_{\substack{y \in W \\ y \neq 0}} \frac{y^H B y}{y^H y}$$

$$\leq \max_{V_{k-1}} \min_{\substack{y \in V_{k-1} \\ y \neq 0}} r_B(x) = \lambda_{k-1}(B)$$

Se W ha dimensione maggiore devo stare considerando tutti i suoi sottospazi, quindi il max è ancor più grande.

Per concludere applico quanto dimostrato a $-A$ e $-B$:

$$-\lambda_n(A) \geq -\lambda_{n-1}(A) \geq \dots \geq -\lambda_1(A)$$

$$-\lambda_{n-1}(B) \geq -\lambda_{n-2}(B) \geq \dots \geq -\lambda_1(B)$$

Ho che $\lambda_{k-1}(-B) \geq \lambda_k(-A)$ $\qquad k = 2, \dots, n$

$\lambda_{k-1}(-B) = -\lambda_{(n-1)-(k-1)+1}(B) = -\lambda_{n-k+1}(B)$ $\qquad -\lambda_{n-k+1}(A) = \lambda_{n-k+1}(A)$

Quindi ribattendo $i = n-k+1$ ho trovato $\lambda_i(A) \geq \lambda_i(B)$ $i = 1, \dots, n-1$ ■

TEO: $A, B \in \mathbb{C}^{n\times n}$ Hermitiane con $B = A + \sigma \underbrace{u u^H}_{\text{Diade}}$ correzione di rango 1, $u \in \mathbb{C}^n$, $\sigma \geq 0$

$$\Rightarrow \lambda_1(A) + \sigma u^H u \geq \lambda_1(B) \geq \lambda_1(A) \geq \dots \geq \lambda_{n-1}(A) \geq \lambda_n(B) \geq \lambda_n(A)$$

dim: se $u = 0$ è ovvio dato che $A = B$. Sia $u \neq 0$ vogliamo mostrare che

$$\lambda_i(B) \geq \lambda_i(A) \qquad i = 1 \dots n$$

Per il TEO CF (1) ho che per $k = 1 \dots n$ $\exists Z_k \subseteq \mathbb{C}^n$ sott.: $\dim Z_k = k$ t.c.

$$\lambda_{n-k+1}(B) = \max_{\substack{x \in Z_k \\ x \neq 0}} r_B(x) = \max_{\substack{x \in Z_k \\ x \neq 0}} \left(\frac{x^H A x}{x^H x} + \sigma \underbrace{\frac{x^H u u^H x}{x^H x}}_{\geq 0} \right)$$

$$\geq \max_{\substack{x \neq 0 \\ x \in Z_k}} \frac{x^H A x}{x^H x} \geq \min_{V_k} \max_{\substack{x \neq 0 \\ x \in V_k}} r_A(x) = \lambda_{n-k+1}(A) \quad \text{dk}$$

(*)

Sempre dalla (1) del Teo CF ho che $\exists W_k \subseteq \mathbb{C}^n$, $\dim W_k = k$ t.c.

$$\lambda_{n-k+1}(A) = \max_{\substack{x \in W_k \\ x \neq 0}} r_A(x)$$

i.e. W_k è quello per cui vale l'= qui (*)

Sia ora $S = \langle u \rangle$ e consideriamo $T = W_k \cap S^\perp$ e prendiamo $x \in T$, ovviamente ho che $x^H u = 0$ (dato che $x \in S^\perp$)

$$x^H B x = x^H A x + \sigma \underbrace{x^H u}_{=0} \underbrace{u^H x}_{=0} = x^H A x \qquad \forall x \in T.$$

Allora abbiamo:

$$\lambda_{n-k+1}(A) \geq \max_{\substack{x \in T \\ x \neq 0}} r_A(x) = \max_{\substack{x \in T \\ x \neq 0}} r_B(x) \quad (**)$$

perché mi sto restringendo a T

Abbiamo che: $\dim S = 1 \longrightarrow \dim S^\perp = n-1$ e quindi

$$\dim(T) = \dim(W_k \cap S^\perp) \geq \max\{0, \dim W_k + \dim S^\perp - n\}$$

$$= \max\{0, k + n - 1 - n\} = k - 1$$

Quindi da (**) posso dire che

$$\lambda_{n-k+1}(A) \geq \min_{V_{k-1}} \max_{\substack{x \neq 0 \\ x \in V_{k-1}}} r_B(x) = \lambda_{n-k+2}(B)$$

↑ perché T ha $\dim \geq 0$ e quindi posso considerare tutti i suoi sottospazi e poi ho $\geq$ perché diminuisco il min

Ora mi rimangono da considerare i λ_1.

$$\lambda_1(B) \overset{CF}{=} \max_{x \neq 0} r_B(x) = \max_{x \neq 0} \left[\frac{x^H A x}{x^H x} + \sigma \frac{|x^H u|^2}{x^H x} \right]$$

$$\leq \max_{x \neq 0} r_A(x) + \sigma u^H u \overset{CF}{=} \lambda_1(A) + \sigma u^H u$$

Cauchy Schwarz

■

GENERALIZZAZIONE: correzione di rango s $B = A + \sum_{i=1}^{s} \sigma_i u_i u_i^H$

o analogamente al teo precedente $B = U^H A U$ con $U \in \mathbb{C}^{n\times m}$ con $m \leq n-1$

(e le idee valgono uguale ...)